

# ソフトウェアの欠陥予防

堀 内 明

## 1. はじめに

家電製品や自動車など多くの製品にはソフトウェアが組み込まれている。水道や交通などの社会インフラもソフトウェアに依存している。このようにソフトウェアへの依存度が高まるにつれ、ソフトウェアの品質問題が多発し、社会問題化している。

携帯電話には、電子メール、画像処理、動画再生、住所録管理など多くの機能があり、組込みソフトウェアによって実現されている。自動車にも、多くの組込みソフトウェアが搭載されている。数多くの車載コンピュータが協調して動作し、燃費や安全性の向上を目指している。

組込みソフトウェアに欠陥があるとリコールが発生するケースが多い。携帯電話、自動車、医療機器など多くの製品においてリコールが発生している。2001年に発生した携帯電話の回収は、大きく報道された。自動車ではエンジンやブレーキに関する不具合によりリコールが発生している。

組込みシステムは、機構部分、電子部品、ソフトウェアからなるシステムであるが、総務省平成16年度統計調査によると、それらを担う技術者数は約38万人と推定されている。そのなかで、約46%にあたる約17万人が組込みソフトウェア技術者である。開発費をみると、組込みソフトウェアの開発費は全開発費の約41%を占める。技術者数、開発費などのコスト面で約4割を占める組込みソフトウェアであるが、製品出荷後

の不具合の原因においても不具合件数の約 34%をソフトウェアが占めている。ハードウェアの不具合が約 23%、製品仕様の不具合が約 22%、その他の不具合が約 21%となっており、製品出荷後の不具合の原因のトップとなっている。

製品の機能が組込みソフトウェアによって実現されることにともない、ソフトウェアの設計開発の特性が製品の設計開発プロセスの中に絡んできたため、人材育成、品質管理等の面で改革が必要となってきたのである。組込みソフトウェアの不具合は、世界規模で多くの利用者に損害を与えることになり、製造メーカーの責任は厳しく問われている。

ソフトウェアのミスは、ソフトウェア開発工程のさまざまな領域で発生する。コード記述時のミスが代表的であるが、設計段階、プロジェクトマネジメントなどにおいても発生する。ミスが発生するのは不可避であり、そのいくつかがソフトウェアの欠陥に結びついている。このような認識から、ソフトウェア開発者は、欠陥を取り除くことに集中し、利用者に影響がでないように動作テストに労力を投入している。

しかし、欠陥予防で重視することは、より効率のよい方法によりソフトウェアの欠陥を取り除くことである。欠陥予防とは、ソフトウェアに欠陥が混入した後、動作テストにより欠陥を検出するだけではなく、欠陥が混入することを防ぐことに焦点を合わせている。さまざまな技法によりソフトウェア開発のプロセスを改善すれば、ソフトウェアの欠陥を減少させることができるのである。

動作テストにより、多くの欠陥を検出できる可能性があり、さまざまな技法も用意されている。しかし、この方法では存在する欠陥を特定することはできるが、他でも発生しうる類似の欠陥を予防することはできない。

不具合が発生する前に欠陥混入を予防するためには、ソフトウェアレビュー、ソフトウェアインスペクションなど、プログラムが動作しない

段階でソフトウェアの欠陥、不具合、問題を発見するための静的解析が必要である。静的解析はソフトウェアテストに先立って実施される。ソフトウェアレビュー、ソフトウェアインスペクション、およびその周辺技術のことを欠陥予防<sup>1)</sup>と呼んでいる。ウォータフォールモデル型の開発では、上流工程において実施されるため、欠陥予防は上流品質向上活動と位置づけられている。

## 2. 組込みソフトウェアに関する不具合発生事例

組込みソフトウェアに関しては、医療機器、家電、自動車において多くの不具合が発生している。公表された不具合事例を以下に示す。

### (1)不具合事例 1<sup>2)</sup>

株式会社ジェイ・エム・エスは、「JMS 個人用透析装置」について、透析前洗浄時に実施される自己診断が開始されず、装置が停止したままになっていたとの情報を医療機関より受領した。発生時の状況を確認したところ、ソフトウェアの不備により特定の状態で停電が発生し、その後復旧して上記事象が発生することが判明したため、同社は改修を実施するとした。

### (2)不具合事例 2<sup>3)</sup>

ゼオンメディカル株式会社が出荷した「ゼメックス IABP コンソール 908」の使用中に、断続的なアラーム音が発生してモニタ上に CPU の異常を示す表示がなされ、駆動が停止したとの報告を受領した。調査の結果、アラーム信号を伝達するソフトウェアに起因して、使用されていたバルーンカテーテル内圧力の異常を検知した後のアラームが CPU の異常を示すアラームへ移行したことが原因と判明したため、同社は回収を行うこととした。

### (3)不具合事例 3<sup>4)</sup>

日本シグマックス株式会社は、「ミネライザー」について、納入先医療機関より「骨密度の測定結果が年代別の平均値を下回ることが少ない」との申し出を受け、当該機器の内部データと実際に測定されたデータの検証を行ったところ、被験者の骨密度を判定するためのデータベースに使用した演算式と実際の測定結果を出力する際に使用する演算式に相違があることを確認した。骨密度判定の基準値と測定結果が同期していないため、被験者の骨密度の測定結果が不正確に出力される傾向があるとして、同社は改修を行うこととした。

### (4)不具合事例 4<sup>5)</sup>

シーメンス旭メディテック株式会社が販売した「オンコア インプレッション プラスシステム」において、モニター画面に表示されたポータル画像に対して操作を行った場合、ポータル画像が視覚的に変化することがあり、視覚的に変化したポータル画像の修正を行うと、表示されているオフセット値が不正確となる場合のあることが判明した。不正確なオフセット値に基づいて患者の位置決めを行うと、誤った部位に対して放射線の照射が行われるため、同社は改修を実施するとした。

### (5)不具合事例 5<sup>6)</sup>

KDDI 株式会社は、「CA003」について以下の不具合を改善するため、最新のソフトウェアの提供を開始した。

- ① カメラ撮影時に、電源がリセットする場合がある。
- ② 音声通話時に相手の声が聞きとりにくい場合がある。

### (6)不具合事例 6<sup>7)</sup>

東芝メディカルシステムズ株式会社は、超音波診断装置「APLIO

ARTIDIA SSH-880 CV」において FLEX-M モードおよびアノテーション機能使用時に以下の問題が発生することが判明したため、改修を実施するとした。

- ① FLEX-M モードで保存した画像を読み出し、Exam Review 上で計測すると、Mモード画像上の計測において誤った値が表示される場合がある。
- ② アノテーション機能にて、タッチパネル上のスイッチを押すと、スイッチ名と異なるコメントが画面上に表示される。

#### (7)不具合事例 7<sup>8)</sup>

GE ヘルスケア・ジャパン株式会社が出荷した「汎用超音波画像診断装置」において、リニアプローブを使用し、2 D画像でCRI をオンにして、パルスドプラモードをアップデートモードに設定した場合に、ドプラの角度補正を使用すると、流速スケールが正しく表示されず、30～60%の範囲で流速が過小評価されることが判明した。同社は改修を実施するとした。

#### (8)不具合事例 8<sup>9)</sup>

「生化学自動分析装置 CA-400 plus」は遠心分離した全血から HbA1c%を測定する機能において、Hb と HbA1c を測定し、その結果を演算して HbA1c%を出力するが、Hb または HbA1c が異常値となった場合に測定結果を出力すべきではないところ、正常値として出力してしまうという不具合が判明した。製造・販売元の古野電気株式会社は改修をするとした。

#### (9)不具合事例 9<sup>10)</sup>

東芝メディカルシステムズ株式会社は、超音波診断装置「XARIOXG

SSA-680 A」において、体腔内プローブを用いたボリューム画像およびボリューム画像から生成される断面像が左右反転するという問題が判明したため、改修を実施するとした。

#### (10)不具合事例 10<sup>11)</sup>

クラリオン株式会社は、パーソナル ナビゲーション デバイス「DTR-P 7 DT」の一部ロットにおいて、製造工程上のミスにより地点登録等の際に漢字変換ができない不具合のあることが判明したとして、ソフトウェアのアップデートを行うよう呼びかけている。

#### (11)不具合事例 11<sup>12)</sup>

台湾において、米国 Haemonetics 社製「Mobile Collection System plus Model 9000」使用中に、ACD ポンプが故障により誤動作を起こし、供血者に対して一過的に ACD 投与され、中程度のクエン酸中毒症状を起こすという事例が発生した。装置のハードウェアの安全機構である安全ボードは、本来ポンプ停止中の返血時にはポンプの動作を監視しておらず、ポンプの誤動作を止められなかったことから、同社は同系列の装置である国内の CCS を対象として、停止中のポンプを監視するよう安全ボードのポンプエンコードのソフトウェア改良を行うこととした。

#### (12)不具合事例 12<sup>13)</sup>

東芝メディカルシステムズ株式会社は、「東芝スキャナ Aquilion TSX-101 A」のオプションソフトウェアである脳血流解析システムにおいて、ソフトウェアの問題により正しい解析結果が得られないという問題があることが判明したため、改修を実施するとした。

**(13)不具合事例 13<sup>14)</sup>**

KTM JAPAN 株式会社は、「690 ENDURO-R」について、速度計のソフトウェアが不適切なために、速度計の配線またはバッテリーの配線を外す作業を行った場合、速度計に実際の速度より速い数値が表示される恐れがあるとして、国土交通省にリコールを届け出た。

**(14)不具合事例 14<sup>15)</sup>**

GE 横河は、特定のソフトウェアのバージョンにおいて、ソフトウェアの不具合のため、特定のパルスシーケンスを使用した場合、位相エンコードの方向設定によっては得られる画像が左右方向に反転し、左右方向を表示する付帯情報とも一致しない場合のあることが確認されたため、改修を行った。

**(15)不具合事例 15<sup>16)</sup>**

富士フィルム株式会社は、1 件前に処理した画像データがコンソールに誤送信され、処理待ちのメニューに入力・表示されるという不具合が発生したため、ソフトウェアを改修した。この不具合は、二つの要因が重なった場合に限り、極めて稀に発生する。

**(16)不具合事例 16<sup>17)</sup>**

米 Segway は、同社が発売した乗用電動 2 輪車「Segway Personal Transporter」を、全数リコールすると発表した。ソフトウェアを改修する。リコール対象の Segway PT は、ソフトウェアの不具合により、走行中にタイヤが逆回転することがあり、乗員が落下するおそれがある。この現象は、スピード制限機能が働き Segway PT が後ろに傾いたときに、乗員がいったん降り、すぐに乗り直したときに起こる。

**(17)不具合事例 17<sup>18)</sup>**

エンジン制御コンピュータのプログラムが不適切なため、高温多湿かつエアコン作動中に、低速走行からアクセルペダルを放し減速すると、エンジンが停止することがある。フォルクス ワゲン グループ ジャパン株式会社は、対象となるプログラムを書き換えることとした。

**(18)不具合事例 18<sup>19)</sup>**

ドレーゲル・メディカルジャパン株式会社は、ソフトウェアバージョン 1.10 がインストールされている「オキシログ 3000」について、ソフトウェアの不具合により特定の設定にした場合にのみ、約 5 秒間換気が中断され、気道内圧がゼロまで低下してアラームを発生する可能性があることが判明したため、修正済みソフトウェアによる改修をおこなった。

**(19)不具合事例 19<sup>20)</sup>**

日本電気株式会社は、「IP 電話対応機器」に関して、ソフトウェアの一部に不具合があり、電源投入後約 6 年 9 ヶ月の間連続して利用すると、電話の発着信ができなくなることが判明したため、ソフトウェアのバージョンアップをお願いしている。

**(20)不具合事例 20<sup>21)</sup>**

ソフトバンクモバイル株式会社は、下記の不具合改善のため、ソフトウェアアップデートのお願いをしている。

- ① 電源を入れた際、ごく稀に「SoftBank」ロゴ画面表示のまま、待ち受け状態にならない場合がある。
- ② 操作していないにもかかわらず、携帯電話機の使用時間が短くなる場合がある。



**(21)不具合事例 21<sup>22)</sup>**

株式会社秋田ケーブルテレビは、下記の不具合改善のため、新バージョンのソフトウェアダウンロードをお願いしている。

- ① 予約録画が実行されないことがある。
- ② 予約録画で録画開始後、時間が経過するとブラックアウトが発生することがある。
- ③ 電源オフからの予約実行で、失敗することがある。

**(22)不具合事例 22<sup>23)</sup>**

株式会社フィリップスエレクトロニクスジャパンは、装置付属「リモートコントロール」での操作に限り、システムが停止する可能性のあることが判明したため、ソフトウェアの改修を行った。

**(23)不具合事例 23<sup>24)</sup>**

ベックマン・コールター・バイオメディカル株式会社は、自動輸血検査装置において、複数のエラー条件が同時に発生した場合、装置の停止直前に測光を終了したプレートの1検体目の測定結果が、次に測光部に移動してきたプレートの1検体目の測定結果で上書きされてしまうという不具合が判明したため、改修を行った。

### **3. 組込みソフトウェアの高品質化**

近年、組込みソフトウェアの品質が大きくクローズアップされる機会が増えている。組込みソフトウェアが搭載される組込みシステムの一部は、交通機器や医療機器などのように人命に直接かわるため、高信頼性を要求されている。このようなシステムにおける組込みソフトウェアの品質<sup>25)</sup>は極めて重要な意味を持っている。

人命には直接かわらない家電機器のような製品であっても、低品質

の製品は消費者に受け入れられないので、企業活動への影響は多大である。このように組込みソフトウェアの品質は、製造者と利用者の両方にとって重要な要素となっている。

### 3.1 ソフトウェア品質の概念

ソフトウェア品質には、2種類<sup>26)</sup>の概念が存在する。1つはプロダクト品質と呼ばれるものであり、他の1つはプロセス品質と呼ばれるものである。プロダクト品質とは、開発されたソフトウェアや開発途中での中間成果物に関する品質である。プロセス品質とは、ソフトウェア開発を進める際の開発作業の質に関するものである。

プロダクト品質としては、ソフトウェア製品の機能性、信頼性、使用性、効率性、保守性、移植性という6種類の特性がある。プロセス品質としては、ソフトウェア開発プロセスの成熟度<sup>27)</sup>が相当する。

### 3.2 高品質化のための技術

組込みソフトウェア開発においては、上流工程の作業の中で、ソフトウェアのみを考えるだけでは不十分である。これらのソフトウェアと密接な関係にあるハードウェアの仕様も考慮しなくてはならない。組込みソフトウェアの領域においてもモデリング作業が重要である。モデリングの道具としては、UMLを利用することができる。組込みソフトウェアのモデリングを行う場合、関連するハードウェアをどの程度考慮するかについては、ソフトウェア設計者に依存している。

品質を向上させるための方法としては、ソフトウェアコンポーネントを利用することも必要である。ビジネスアプリケーションの領域ではさまざまなソフトウェアコンポーネントが普及しており、これらのコンポーネントを利用した開発が常識となっている。しかし、組込みソフトウェアの領域においては、ソフトウェア部品を活用して効率的にソフト

ウェアを開発する形態が十分にはできていない。今後は、組込みソフトウェア向けのソフトウェア部品の整備と流通の促進が重要である。

組込みソフトウェアの開発<sup>28)</sup>においては、テスト作業が一番難しい。組込みソフトウェアは、開発環境と実行環境とが異なるクロス環境下で開発される場合が多い。このため、開発環境におけるテストでは問題がなかったとしても、実行環境においてはハードウェアとのインタラクションが生じるので、開発環境においては予期できなかった要因により、動作が不安定になる場合がある。最近の組込みソフトウェアは規模が大きくなってきているため、テスト項目の数も増加している。しかし、開発環境下でテスト可能な項目には限界があり、動作テストだけではなく、ソフトウェアの静的な検証方法を活用し、機能や動作を保証することが必要である。

静的な検証方法としては、ソフトウェアレビューやソフトウェアインスペクションがある。ソフトウェアレビューやソフトウェアインスペクションは、ソフトウェアの中間的な成果物を対象とし、作成担当者以外の者による目視によって欠陥を検出し、指摘する活動である。

#### 4. ソフトウェアの品質目標

高品質のソフトウェアを作成するための技法としては、ソフトウェア設計の最適化、レビューや動作テストの充実、開発プロセスの改善などさまざまな技法や考え方が導入されている。ソフトウェアの品質に関しては、さまざまな定義や測定方法が提案されている。

しかし、従来のソフトウェア品質に関する技法においては、レビューや動作テストの到達目標、求められている品質のレベル、達成するための技法の選択などについての具体的な指針が示されているわけではない。その結果、開発現場においては、明確な品質目標<sup>29)</sup>にもとずいてソフトウェア開発を行うことができず、目標値に到達するための品質の定

量的制御という考え方が実践されていない。それによりソフトウェアの不具合や障害が発生し続けている。

### (1)品質指標の概念

世の中に存在する「もの」や「作業」の特質を表現する方法として指標という概念があるが、ソフトウェアにも特質がありその特質を制御する概念として、品質指標を導入することができる。ハードウェアと比較してソフトウェアは、製品から分離して観察することが不可能であり、計測が難しい対象物である。ソフトウェアは、ソースコードや設計書という中間的な成果物の形でのみ対象物としてとらえることができる。したがって、ソフトウェアを計測対象とすると、極めてとらえることが難しい対象物である。

しかし、世の中のさまざまな組込みシステムにおいては、不具合が発生したり、利用者が使いにくい製品が数多く存在している。ソフトウェアは、計測の対象物としては極めて難しいものではあるが、他の有形物と同様に品質という概念を適用することが可能である。ソフトウェアの場合、最終成果物としての品質を測るだけでなく、開発過程の作業についても評価する必要がある。

最終成果物を得る過程において行った作業の質を測る指標のことを、プロセス品質評価指標といい、最終成果物として得られたソフトウェアの質を測る指標のことを、プロダクト品質評価指標と呼んでいる。

プロセス品質評価指標の対象となる作業項目としては、要求分析、システム設計、プログラム設計、コーディング、プログラムテスト、システムテストなどがある。プロダクト品質評価指標の対象物としては、要求定義書、システム設計書、プログラム設計書、コード、プログラムテスト仕様書、システムテスト仕様書などがある。

## (2) プロセス品質評価指標

プロセス品質評価指標<sup>30)</sup>とは、ソフトウェア開発の過程で実施される作業を計測し評価するための指標である。品質の高いソフトウェアを作成するためには、開発工程において品質面の点検や確認などの作業が適切に行われる必要がある。

プロセス品質評価指標の対象となる作業は、とらえることが難しい。一般的には、作業が適切に実施されているか、作業が十分に行われているかが評価の観点になる。作業の適切さや十分さについては、作業充当率や作業実施率により評価することができる。

### ① 作業充当率

作業充当率とは、ソフトウェア開発の過程で作成される中間成果物や最終成果物のレビューに要した工数を当該プロセスに要した作業工数で除した値である。これにより、作業工数の十分さを評価することができる。要求仕様、システム設計、コード作成、テストについてそれぞれのレビューに要した作業工数を、それぞれのプロセスの全作業工数で除することにより求められる。たとえば、システム設計レビューの作業充当率は、システム設計レビューに要した作業工数を、システム設計に要した全作業工数で割った値である。

### ② 作業実施率

作業実施率とは、ソフトウェア開発の過程で作成される中間成果物や最終成果物のレビューに要した工数をソフトウェアの規模を示す量で除した値である。これにより、作業の質的な十分さを評価することができる。要求仕様、システム設計、コード作成、テストについてそれぞれのレビューに要した作業工数を、ソフトウェア全体の規模で除することにより求められる。ソフトウェアの規模としては、ソースコードの全行数を使用する。たとえば、システム設計レビューの作業実施率は、システム設計レビューに要した作業工数を、ソースコードの全

行数で割った値である。

### (3)プロダクト品質評価指標

ソフトウェア開発の過程においては、設計書、ソースコード、テスト仕様書などさまざまな成果物が作成される。プロダクト品質評価指標は、このように開発の過程において作成される中間成果物や最終成果物の質を評価するための指標である。成果物の持つ特質の中で、ページ数、行数、不具合数などの量的側面を計測し、品質に結びつく要素が反映されているかどうかを評価する。プロダクト品質評価指標は、評価する対象により、ドキュメント、ソースコード、テストの3つに分けることができる。

#### ①ドキュメント品質評価指標

ドキュメント品質評価指標とは、ソフトウェア開発の過程において作成される中間成果物や最終成果物である仕様書、設計書などの文書の品質を評価するための指標である。指標としては、ドキュメントボリューム品質評価指標とドキュメントバランス品質評価指標がある。

ドキュメントボリューム品質評価指標とは、ドキュメントが品質という観点で適切であるかどうかを測る指標として、そのドキュメントの量（文字や図表）を計測する指標である。ドキュメントバランス品質評価指標とは、ドキュメントの量には問題がないとしても、重要な項目が欠けていれば、開発に支障が生じるので、各ドキュメントに記載すべき項目が、実際にはどの程度記載されているかを、割合で示すものである。

#### ②コード品質評価指標

コード品質評価指標とは、ソフトウェア実装時の最終成果物であるソースコードを直接観察し、品質を評価する指標である。ソースコードの量について計測する方法としては、コメントを除いた行数を計測

する方法とコメントを含めて計測する方法とがある。ソースコードの行数は、開発者の記述方法により差異が生じるので、コーディング基準を作成して均一化を図らなければならない。それにより、ソースコード品質評価指標の有効性が保証されるのである。コード品質評価指標は、コードボリューム品質評価指標とコード特性品質評価指標とに分けられる。

ソースコードが品質という観点からみて、適切であるかどうかを測る目安として、ソースコードの量が適切であるかどうか一つの指標となる。単位あたりのソースコード行数が、一定規模以上になると、保守性や理解度などに悪影響を与えるからである。コードボリューム品質評価指標としては、ソースファイルに記述されたソースコードの量、ソースコードを構成する関数あたりのソースコードの量がある。

ソースコードが量的に一定の値以下であったとしても、記述方法に偏りがあると、保守性や可読性の点で悪影響が生じることとなり、ソースコードの品質低下につながるのである。たとえば、コメント行は、レビュー担当者や保守担当者に対して、当該ソースコードの機能についての貴重な情報提供手段となっているので、適切な記述が要求される。コメント行記述率を評価することにより、情報不足をある程度判断することができるのである。

### ③テスト品質評価指標

テスト品質評価指標とは、最終成果物であるオブジェクトコードを動作させるテストが十分であるかどうかを評価するための指標である。テスト品質評価指標は、テスト十分性品質評価指標と動作完全性品質評価指標とに分けられる。テスト十分性品質評価指標とは、テストのためのデータが十分であるかどうかという指標であり、テストの十分性を測る指標の一つである。動作完全性品質評価指標とは、テスト作業後のソフトウェアの品質がどの程度安定しているかを示す指標

である。検出された不具合がどの程度修正されているかを評価することにより、出荷後のリスクを評価することができるのである。

## 5. ソフトウェアテストの改善

コンピュータ初期の頃のテストは、プログラマによるプログラムの動作確認作業という位置づけであった。そのため、テストケースが十分な考慮の上設計されていたわけではなかった。また、ソフトウェア工学のような技術分野として認知されていなかった。その後、パラダイムシフトが起こり、不具合を発見するためには適切なテスト設計が必要であるという考え方が提起され、テストが重要な技術分野として認識されるようになった。

### 5.1 テスト設計

テスト作業において不具合を検出するためには、さまざまな条件下でテストの対象となっているソフトウェアを動作させる必要がある。条件としては、入力値、動作環境、タイミングなどがあり、それぞれの条件の組み合わせのことをテストケースと呼んでいる。テストケースをシステムティックに作成することをテスト設計と呼んでいる。適切なテスト設計<sup>31)</sup>を行うことにより、少ないテストケースでも多くの不具合を発見することが可能である。しかし、単なる思いつきによりテストケースを作成すると、多くの不具合を見逃してしまう危険性がある。

テスト対象のある側面に注目してテスト設計する技術のことをテスト技術と呼んでいる。テストにおいて多くの不具合を発見できるようなテスト設計を可能にするためには、単一のテスト技術だけでは不十分である。

テスト設計の基本は網羅である。ソフトウェアを網羅的にテストできるのであれば、すべての不具合を発見することができ、品質リスクはな



くなる。したがって、網羅的なテスト設計を行い、網羅性を担保することが必要となる。しかし、実際には工数の問題があり、網羅的テストは不可能である。間引きによってテストケースを削減しなければならない。削減したテストケースにおいて発見できる不具合による品質リスクを一定の値以下に抑えることが必要である。間引きの方法としては、グルーピングがある。グルーピングにより品質リスクを増加させずに間引きすることができる。

テスト設計の基本には、ピンポイント型テスト技術がある。不具合を発見できるようなテストケースだけをピンポイントに設計しテストできるならば、網羅的にテストする必要はない。ピンポイントにテストを行うと、品質リスクを増加させずにテストケースを間引きできる。境界値付近は不具合が発生する危険が多いので、境界値テストはピンポイント型テストの一例である。

テスト設計の作業を行っている際に不具合を発見する場合もある。テスト設計であれば、ソフトウェアが完成している必要がないため、上流工程において行うことができる。したがって、不具合を予防することができるのである。

## 5.2 テスト技術

質の高いテスト設計を行うためには、多くのテスト技術を把握しておく必要がある。テスト技術の多くは、テスト作業の前にテスト設計を行っておく確定的テストである。しかし、テスト技術の中には、テスト作業の前にテスト設計をおこなわない非確定的テストもある。

確定的テストは、テスト対象の一側面に注目してテスト設計を行うテスト技術である。したがって、ある側面でのテスト設計が不十分であっても、他の側面によるテスト設計で補うことが可能である。

テスト作業は下流工程で実施されるので、上流工程で利用されている

開発技術に対応したテスト技術を適用した方がより効果的である。上流工程において利用された開発技術が構造化分析や構造化設計であれば、伝統的なテスト技術を適用できる。オブジェクト指向の方法により開発されたテスト対象の場合には、UML 図を網羅するテストなどが必要となる。

テスト技術はソフトウェアの領域によっても異なる。組込みシステムにおいては、ハードウェアとのインタフェースの不整合についてのテスト、リアルタイム制約についてのテスト、ハードウェア故障時の対応についてのテストなどが必要となる。

企業の情報システムにおいては、業務フローをモデル化し、網羅的にテストすることが必要である。データモデルを考慮して、データ間の関係に注目したテストも必要である。

パッケージソフトウェアにおいては、利用者の利用形態を把握したテストが必要である。そのためには、利用するシナリオを網羅するテストや使いやすさを考慮したテストを行わなければならない。また、同時に動作する他のソフトウェアや周辺機器との相性を評価する必要がある。

## 6. ソフトウェアの静的な解析

ソフトウェアの品質を向上させるためには、評価による不具合の検出と修正を行うことが必要である。このための方法として、ソフトウェアテストやソフトウェアインスペクションがある。ソフトウェアテストは、対象となるソフトウェアが実行可能でなければ実施できないが、ソフトウェアインスペクションはソフトウェアの中間成果物を対象として実施するものである。その点から、ソフトウェアの静的解析方法として位置づけられている。

ソフトウェアインスペクションは、不具合を早期に検出し修正することができるので、やり直しの手間や時間が動作テストと比較して少なく

て済むのである。設計段階において欠陥が混入し、動作テストにおいて欠陥を発見しようとする場合、コード作成や単体テストが完了した後の結合テストにおいて設計上の欠陥が発見され修正される。この場合には、欠陥を含んだ設計書に基づいて作成されたソースコードが修正され、修正後のソースコードに対して、再度単体テストを実施しなければならない。設計時のインスペクションにより欠陥を発見できれば、欠陥を含んだ設計書に基づいたソースコードの作成や単体テストをしないで済むのである。

設計書などのドキュメントを作成した者以外で開発に従事しているメンバがインスペクションの担当者となり、開発者としての知識を活用しながらインスペクションを行う。インスペクション担当者は、設計書の作成担当者である必要はなく、ソースコード作成者やテスト担当者である場合もある。この方法の利点は、対象となるソフトウェアを熟知しているメンバによって実施されるため、対象となるソフトウェアに対する理解に必要な時間を省ける点にある。しかし、利用者側からの視点ではなく、開発者側からの視点による欠陥の指摘なので、ユーザビリティの欠陥などが指摘されにくいという点もある。この点を克服する方法として、利用者をインスペクション担当者とする方法がある。それにより、利用者が期待しているソフトウェアが開発されているかどうかを確認することができる。インスペクション担当者である利用者は、ソフトウェアが利用者が求めるものになっているかどうかという視点で指摘する。この方法の欠点は、利用者がソフトウェアの内部についての理解をするまでに時間がかかる点である。開発当事者ではないが、ソフトウェアに関して知識を有している第三者をインスペクション担当者とすることも有益である。第三者により、開発者とは異なった視点からのインスペクションを実施できる利点がある。

インスペクションプロセスを改善するには、リーディング手法の改善

が効果的である。リーディング手法の相違によって、欠陥検出の有効性に関して差のあることが研究によって示されている。リーディング手法に関する研究によれば、アドホックなリーディング手法と体系的なリーディング手法とを比較すると、体系的なリーディング手法の方がより効果的であることが示されている。チェックリストに基づくリーディングの場合、過去に生じた欠陥に関する経験的知識を網羅的に適用できるという利点がある反面、チェックリストの量が増加するため、すべての項目を確認するのが容易ではないという欠点がある。

## 7. 開発プロセスの改善

あらゆるソフトウェアモデルの中で、最初に提唱されたモデルが、CMM である。CMM の目的は、開発に必要なプロセスを対象として、組織の能力を評価することであった。モデルを実践した結果、プロセス改善<sup>32)</sup>にも有益であると認識されるようになった。CMM は、モデルに必要な各要素に関して、組織が持つプロセスの成熟度を表すものである。CMM は、内容が豊富であり、複雑な改善モデルである。CMM においては、プロセスの成熟度を5つのレベルに分けており、それぞれのレベルには、目的がある。各レベルは、一つ下のレベルにおいて修得した能力を基礎として、その上に成り立っている。

レベル1は、初期または特定の目的のために設計されたプロセスの使用を意図したものである。レベル1のプロセスが実装されると、プロセスの理論的根拠は無視され、他のプロジェクトに不適切なプロセスを強いる恐れがある。レベル1の組織の問題は、作業プロセスを考えたり、不適切なプロセスを使用することに時間を費やさなければならないことである。

レベル2は、反復可能なプロセスを持つ組織である。あるプロジェクトにおいて修得されたスキルが、そのまま次のプロジェクトに移行可能

であることを意味している。このレベルの目的は、プロジェクト計画における組織の能力を理解することである。要求管理や構成コントロールなどのプロジェクト管理上重要な問題の多くは、事前に計画できるのである。

レベル3は、定義されたレベルと呼ばれている。レベル3においては、組織全体で使用する標準化されたプロセスを組織が開発する。標準化されたプロセスを使用することによって、各プロジェクトは学習したことを活用し、それを他のプロジェクトへ伝えることができる。各プロジェクトは、プロジェクトにとって有益である標準プロセスを使用する。ベストプラクティスである標準的なマネジメントプロセスを使用すれば、複数のプロジェクトを協調させることが可能となる。

レベル4は、管理されたレベルと呼ばれている。レベル4の組織においては、標準的な組織のプロセスによって収集されたデータを用いて、統計的手法によりプロジェクトを管理する。組織が一貫したデータを収集できなければ、統計のプロセスコントロールのような技法を利用することはできないのである。

レベル5は、最適化するレベルと呼ばれている。このレベルの実践は、科学的に有効な方法により標準プロセスを改善することである。このレベルでは、レベル4において得られた情報の活用とコントロールされたベストプラクティスの導入が必要となる。

CMMは、ソフトウェア開発プロセスに容易に適用可能なモデルであるが、より複雑なモデルとしてCMMIモデルがある。CMMIモデルが必要な理由は、ソフトウェア開発の部署とハードウェア開発の部署が異なるような組織をサポートする必要があるからである。CMMIにおいては、モデルに多くの情報を追加している。CMMIは、組織成熟度の判定が可能な段階表現と呼ばれるモデルであり、レベル1からレベル5までの5段階に対応付けられた複数のプロセスが一定以上の能力レベルに達

しているか否かによって、組織成熟度レベルの評価を行う。ソフトウェアのプロセス評価においては、CMMI がデファクト規格として、国際的に活用されている。

## 8. おわりに

ソフトウェアの品質向上のためには、伝統的なソフトウェアテストと並んで、ソフトウェアインスペクションが重要であることを強調した。ソフトウェアインスペクションにより、上流工程において欠陥を除去できるので、品質を高めると同時に開発コストを低減させることが可能である。

ソフトウェアも一つの製品であり、一般的な品質管理の手法を取り入れることも重要である。たとえば、品質管理の基本として、トレーサビリティ<sup>33)</sup>という概念がある。ソフトウェアの開発においてもトレーサビリティを導入することにより、品質向上に寄与できるものとする。しかし、一般のソフトウェア開発において、トレーサビリティが定着しているとはいえない。理由としては、トレーサビリティのためのコストに対して、効果が十分に認識されていないからである。トレーサビリティの効果については、ソフトウェアの開発者だけでなく、管理者や利用者の品質管理に対する理解が不可欠である。

### 注

- 1) 森崎修司「ソフトウェアレビュー/ソフトウェアインスペクションと欠陥予防の現在」『情報処理』、Vol.50, No.5, 2009, p.375
- 2) JMS「個人用透析装置」改修、<http://www.recall-plus.jp/info/11584>, 2010.2.17
- 3) ゼオンメディカル「補助循環用バルーンポンプ駆動装置」改修、<http://www.recall-plus.jp/info/11605>, 2010.2.17

- 4) SIGMAX「超音波骨密度測定装置」改修、<http://www.recall-plus.jp/info/11688>, 2010.2.17
- 5) SIEMENS「線形加速器システム」改修、<http://www.recall-plus.jp/info/11564>, 2010.2.17
- 6) au「CA003 アップデート」お知らせ、<http://www.recall-plus.jp/info/11452>, 2010.2.17
- 7) TOSHIBA「循環器用超音波画像診断装置」改修、<http://www.recall-plus.jp/info/11094>, 2010.2.17
- 8) GE ヘルスケア・ジャパン「汎用超音波画像診断装置」改修、<http://www.recall-plus.jp/info/11064>, 2010.2.17
- 9) FURUNO「移動式ディスクリット方式臨床化学自動分析装置」改修、<http://www.recall-plus.jp/info/11056>, 2010.2.17
- 10) TOSHIBA「汎用超音波診断装置」改修、<http://www.recall-plus.jp/info/10832>, 2010.2.17
- 11) clarion「パーソナル ナビゲーション デバイス DTR-P7DT 漢字変換不具合」お詫び、<http://www.recall-plus.jp/info/10794>, 2010.2.17
- 12) HAEMONETICS「遠心型血液成分分離装置」改修、<http://www.recall-plus.jp/info/10537>, 2010.2.17
- 13) TOSHIBA「全身用X線 CT 診断装置」改修、<http://www.recall-plus.jp/info/10364>, 2010.2.17
- 14) KTM「690 ENDURO-R」リコール、<http://www.recall-plus.jp/info/10152>, 2010.2.17
- 15) GE Healthcare「超電導磁石式全身用 MR 装置」改修、<http://www.recall-plus.jp/info/9765>, 2010.2.17
- 16) FUJIFILM「コンピューテッドラジオグラフ」改修、<http://www.recall-plus.jp/info/9689>, 2010.2.17
- 17) 米 Segway「旧モデルを全数リコール」、<http://techon.nikkeibp.co.jp/article/NEWS/20060915/1211198>, 2010.2.9
- 18) フォルクスワーゲン グループ ジャパン株式会社「サービスキャン

- ペーン 通知書」、<http://www.volkswagen.co.jp/service/recall/>、2010.2.9
- 19) Draegermedical「オキシログ 3000」点検修理、<http://www.recall-plus.jp/info/7278>, 2010.2.18
- 20) NEC「IP 電話対応機器」お知らせ、<http://www.recall-plus.jp/info/8501>, 2010.2.18
- 21) SoftBank「SoftBank6 機種ソフトウェアアップデート」、<http://www.recall-plus.jp/info/8772>, 2010.2.18
- 22) CNA「Panasonic 製 STB アップデート」お知らせ、<http://www.recall-plus.jp/info/8927>, 2010.2.18
- 23) PHILIPS「移動型デジタル式汎用一体型 X 線透視診断装置」改修、<http://www.recall-plus.jp/info/8841>, 2010.2.18
- 24) BECKMAN COULTER「自動輸血検査装置」改修、<http://www.recall-plus.jp/info/11130>, 2010.2.18
- 25) 田丸喜一郎「日本の組込みシステム産業の現状と課題」『情報処理』、Vol. 46, No.7, 2005, p.799
- 26) 独立行政法人情報処理推進機構ソフトウェア・エンジニアリング・センター編著『組込みソフトウェア開発向け品質作り込みガイド』、翔泳社、2008、p.9
- 27) Marc McDonald, Robert Musson, Ross Smith（溝口真理子、依田光江訳・宋雅彦監修）『ソフトウェアの欠陥予防』、日経 BP ソフトプレス、2008、p.26
- 28) 平山雅之「組み込みソフトウェア開発の現状」『情報処理』、Vol.45, No. 7, 2004, p.677
- 29) 片山卓也「高信頼性組み込みソフトウェア開発のための技術課題」『情報処理』、Vol.47, No.5, 2006, p.488
- 30) 松本健一「ソフトウェア管理技術の現状」『情報処理』、Vol.44, No.4, 2003, p.327
- 31) 秋山浩一「組合せテストの設計」『情報処理』、Vol.49, No.2, 2008, p.140



- 32) Neil S. Potter, Mary E. Sakry (富野橋・荒木貞雄訳)『ソフトウェアプロセス改善の基本定石』、共立出版、2005、p.71
- 33) 宇田川佳久「情報システム開発標準におけるトレーサビリティの事例と今後」『情報処理』、Vol.51, No.2, 2010, p.150